

Entering RLL Programs

In This Chapter. . . .

- Entering Simple Ladder Programs
 - Checking for Program Errors
-

Entering Simple Ladder Programs

Purpose of the Examples

This section includes many examples that are intended to help you become familiar with the keystrokes required to enter the most basic DL305 instructions. Once you are familiar with the basic keystrokes, you should use the DL305 User Manual as a reference for the remaining instructions.

Handheld Key Sequences

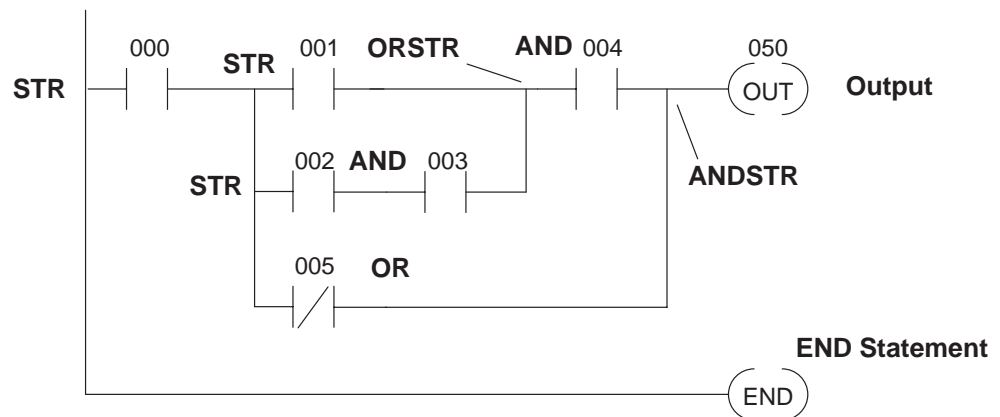
The Handheld buffers all keystrokes until you press the **ENT** key. Then, it automatically checks the instruction to make sure that it has been entered correctly. If the instruction is entered incorrectly an error message will be displayed. See Chapter 6 for a complete listing of error messages.

The Basics

There are a few basic instructions that you must become familiar with to enter programs with the Handheld.

- STR – Stores a normally open element and indicates the beginning of a rung or network.
- AND – Joins one element (such as a contact) in series with another element or group of elements.
- AND STR – Joins a group of elements in series with another group of elements.
- OR – Joins a one element in parallel with a previous element or group of elements.
- ORSTR – Joins parallel branches (each branch must begin with a STR instruction)
- OUT – Each rung must have at least one output (Y, C, or box instruction)
- NOT – used with other instructions to utilize normally closed elements.
- All programs must contain an END statement (automatically provided).

The following diagram shows a typical network and how each of these elements are used.



Starting at Address 0

If you're entering a complete program, you should always start at Address 0. The following example shows the keystrokes required to start at address 0000. (The remaining examples will not show this step, but it is required.)

Start at address 0

SHF NXT

Address Display

<i>0.0.0.0</i>				0	4	0	4
				(AND)	(OUT)	(MCS)	(ADR)
ADDRESS/DATA				1	5	1	5
				(OR)	(TMR)	(MCR)	(SHF)
ON/OFF	RUN	BATT		2	6	2	6
				(STR)	(CNT)	(SET)	(DATA)
PWR CPU				3	7	3	7
				(NOT)	(SR)	(RST)	(REG)

Once you're at address 0, you can start entering a program. After you start entering the program, the Handheld automatically increments to the next address after you enter an instruction. You can toggle from the address display to the data display by pressing the NXT key. You can toggle from the data display to the address display by pressing the CLR key. For example, if you start at address 0000 and press NXT, the display changes and shows the instruction type located at address 0000. The following example shows what the display would look like.

Start at address 0, change to data display instead of address display

NXT

<i>End</i>				0	4	0	4
				(AND)	(OUT)	(MCS)	(ADR)
ADDRESS/DATA				1	5	1	5
				(OR)	(TMR)	(MCR)	(SHF)
ON/OFF	RUN	BATT		2	6	2	6
				(STR)	(CNT)	(SET)	(DATA)
PWR CPU				3	7	3	7
				(NOT)	(SR)	(RST)	(REG)

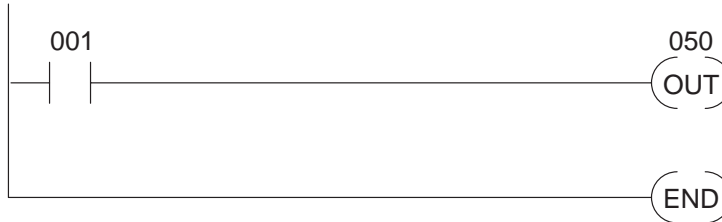
The example keystrokes shown throughout this chapter will indicate which display method is being used. This will make the examples easier to follow. If you prefer a different display you now have the means to change it.

Entering an END Statement

All DL305 programs must have an END statement as the last statement in the program. Whenever you clear the CPU memory, the CPU assumes that *all* memory locations contain an END statement. This means that you do not have to enter an END statement. Just enter your program starting at address 0. You should be aware that if an END statement precedes your ladder logic, the program will not be executed.

Entering Simple Rungs

You use the STR instruction to start rungs that contain both contacts and coils. The following example shows how to enter a single contact and a single output coil. Remember, with the DL305 CPUs, you do not have to enter an END statement with the Handheld Programmer. In the following example, notice that when you enter the output and move to the next address, the END statement is already there.



Enter the contact

STR	SHF	1	ENT
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Data display *before* ENT is pressed

001		0	4	0	4
		(AND)	(OUT)	(MCS)	(ADR)
ADDRESS/DATA		1	5	1	5
		(OR)	(TMR)	(MCR)	(SHF)
ON/OFF	RUN BATT	2	6	2	6
		(STR)	(CNT)	(SET)	(DATA)
PWR	CPU	3	7	3	7
		(NOT)	(SR)	(RST)	(REG)

Enter the output

OUT	SHF	5	0	ENT
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

050		0	4	0	4
		(AND)	(OUT)	(MCS)	(ADR)
ADDRESS/DATA		1	5	1	5
		(OR)	(TMR)	(MCR)	(SHF)
ON/OFF	RUN BATT	2	6	2	6
		(STR)	(CNT)	(SET)	(DATA)
PWR	CPU	3	7	3	7
		(NOT)	(SR)	(RST)	(REG)

Check the next address

NXT
<input type="checkbox"/>

END Statement

End		0	4	0	4
		(AND)	(OUT)	(MCS)	(ADR)
ADDRESS/DATA		1	5	1	5
		(OR)	(TMR)	(MCR)	(SHF)
ON/OFF	RUN BATT	2	6	2	6
		(STR)	(CNT)	(SET)	(DATA)
PWR	CPU	3	7	3	7
		(NOT)	(SR)	(RST)	(REG)

The example shows an input contact and an output coil. Control relays are entered exactly the same as the I/O points.

Entering Normally Closed Elements

Normally closed elements are entered by using the STR and NOT instructions. The following example shows a simple rung with a normally closed contact.



Enter the contact

STR NOT SHF 1 ENT

Enter the output

OUT SHF 5 0 ENT

Data Display before ENT is pressed

001		0	4	0	4	
ADDRESS/DATA		AND	OUT	MCS	ADR	
		1	5	1	5	
		OR	TMR	MCR	SHF	
ON/OFF	RUN	BATT	2	6	2	6
			STR	CNT	SET	DATA
PWR	CPU	3	7	3	7	
		NOT	SR	RST	REG	

050		0	4	0	4	
ADDRESS/DATA		AND	OUT	MCS	ADR	
		1	5	1	5	
		OR	TMR	MCR	SHF	
ON/OFF	RUN	BATT	2	6	2	6
			STR	CNT	SET	DATA
PWR	CPU	3	7	3	7	
		NOT	SR	RST	REG	

Entering Timer/Counter Contacts



Enter the contact

STR TMR SHF 6 0

0 0 ENT

Enter the output

OUT SHF 5 0 ENT

Data Display before ENT is pressed

600		0	4	0	4	
ADDRESS/DATA		AND	OUT	MCS	ADR	
		1	5	1	5	
		OR	TMR	MCR	SHF	
ON/OFF	RUN	BATT	2	6	2	6
			STR	CNT	SET	DATA
PWR	CPU	3	7	3	7	
		NOT	SR	RST	REG	

050		0	4	0	4	
ADDRESS/DATA		AND	OUT	MCS	ADR	
		1	5	1	5	
		OR	TMR	MCR	SHF	
ON/OFF	RUN	BATT	2	6	2	6
			STR	CNT	SET	DATA
PWR	CPU	3	7	3	7	
		NOT	SR	RST	REG	